L4: Entry 1 of 1 File: USPT Nov 16, 1999

DOCUMENT-IDENTIFIER: US 5987506 A
TITLE: Remote access and geographically distributed computers in a globally addressable storage environment

Abstract Text (1):
A computer system employs a globally addressable storage environment that allows a plurality of networked computers to access data by addressing even when the data is stored on a persistent storage device such as a computer hard disk and other traditionally non-addressable data storage devices. The computers can be located on a single computer network or on a plurality of interconnected computer networks such as two local area networks (LANs) coupled by a wide area network (WAN). The globally addressable storage environment allows data to be accessed and shared by and among the various computers on the plurality of networks.

Brief Summary Text (4):
The conventional computer network includes a number of client computers connected together and further connected to a server computer that stores the data and the programs that client computers employ during network operation. This configuration is generally referred to as a client-server network. Typically, each client is a conventional computer system that includes a private main memory, typically a RAM memory, and a persistent storage, typically a hard disk. The server is usually an expensive high end machine that includes a high speed processor unit and a large memory, often having ten to one hundred times more storage than the individual client computers. The clients and server cooperate to share data and services among the different users, and thereby the individual computers appear as a unified distributed system. To this end, the server acts as a central controller that provides through its large memory a central repository of network data, and that distributes services to the individual client computers, generally on an as-available basis. Typically, these services are provided by means of specialized software running on a high speed processor on the server computer.

Brief Summary Text (18):
A computer system according to the invention employs a globally addressable storage environment that allows a plurality of networked computers to access data by addressing even when the data is stored on a persistent storage device such as a computer hard disk and other traditionally non-addressable data storage devices. The computers can be located on a single computer network or on a plurality of interconnected computer networks such as two local area networks (LANs) coupled by a wide area network (WAN). Also, the computers can include remote computers that access the network(s) via a communications adapter (e.g., a modem) and the telephone lines. The globally addressable storage environment allows data to be accessed and shared by such remote computers and among the computers on the plurality of networks.

Detailed Description Text (9):
Still referring to FIG. 1, the system 10 maintains within the addressable shared memory space 20 a structured store of data 28. Each of the nodes 12a-12d can access the addressable shared memory space 20 through the shared memory subsystems 34a-

34d. Each of the shared memory subsystems 34a-34d provides its node with access to the addressable shared memory space 20. The shared memory subsystems 34a-34d coordinate each of the respective node's memory access operations to provide access to the desired data and maintain data coherency within the addressable shared memory space 20. This allows the interconnected nodes 12a-12d to employ the addressable shared memory space 20 as a space for storing and retrieving data. At least a portion of the addressable shared memory space 20 is supported by a physical memory system that provides persistent storage of data. For example, a portion of the addressable shared memory space 20 can be assigned or mapped to one or more hard disk drives that are on the network or associated with one or more of the network nodes 12a-12d as local hard disk storage for those particular nodes. Accordingly, FIG. 1 illustrates that shared memory subsystems provide the network nodes with access to an addressable shared memory space, wherein at least a portion of that space is assigned to at least a portion of one or more of the persistent storage memory devices (e.g., hard disks) to allow the nodes addressably to store and retrieve data to and from the one or more persistent storage memory devices. A preferred embodiment of such an addressable shared memory space described in the commonly-owned U.S. patent application Ser. No. 08/754,481 file Nov. 22, 1996, and incorporated by reference above.

Detailed Description Text (14):
The file system 60 of FIG. 2 differs from known physical and distributed file systems in a variety of ways. In contrast to known physical file systems which map a file organization onto disk blocks, the file system 60 manages the mapping of a directory and file structure onto a distributed addressable shared memory system 20 which has at least a portion of its addressable space mapped or assigned to at least a portion of one or more persistent storage devices (e.g., hard disks) on the network. Unlike known distributed file systems, the file system 60 employs peer nodes, each of which have an incarnation or instance of the same data control program. Also, unlike known file systems generally, the file system 60: maintains data coherence among network nodes; automatically replicates data for redundancy and fault tolerance; automatically and dynamically migrates data to account for varying network usage and traffic patterns; and provides a variety of other advantages and advances, some of which are disclosed in the commonly-owned U.S. patent application Ser. No. 08/754,481 filed Nov. 22, 1996, and incorporated by reference above.

Detailed Description Text (61):
In one aspect, the invention of the above-identified, incorporated-by-reference U.S. patent application can be understood to include computer systems having a addressable shared memory space. The systems can comprise a data network that carries data signals representative of computer readable information a persistent memory device that couples to the data network and that provides persistent data storage, and plural computers that each have an interface that couples to the data network, for accessing the data network to exchange data signals therewith. Moreover, each of the computers can include a shared memory subsystem for mapping a portion of the addressable memory space to a portion of the persistent storage to provide addressable persistent storage for data signals.

Detailed Description Text (63):
The systems can also include a cache system for operating one of the local persistent memory devices as a cache memory for cache storing data signals associated with recently accessed portions of the addressable memory space. Further the system can include a migration controller for selectively moving portions of the addressable memory space between the local persistent memory devices of the plural computers. The migration controller can determine and respond to data access patterns, resource demands or any other suitable criteria or heuristic. Accordingly, the migration controller can balance the loads on the network, and move data to nodes from which it is commonly accessed. The cache controller can be a software program running on a host computer to provide a software managed RAM and

disk cache. The RAM can be any volatile memory including SRAM, DRAM or any other volatile memory. The disk can be any persistent memory including any disk, RAID, tape or other device that provides persistent data storage.

Detailed Description Text (67):
The systems can include additional elements including a paging element for remapping a portion of the addressable memory space between one of the local volatile memory devices and one of the local persistent memory devices; a policy controller for determining a resource available signal representative of storage available on each of the plural computers and, a paging element that remaps the portion of addressable memory space from a memory device of a first computer to a memory device of a second computer, responsive to the resource available signal; and a migration controller for moving portions of addressable memory space between the local volatile memory devices of the plural computers.

Detailed Description Text (74):
In another aspect, the invention of the above-identified, incorporated-by-reference U.S. patent application can be understood as methods for providing a computer system having a addressable shared memory space. The method can include the steps of providing a network for carrying data signals representative of computer readable information, providing a hard-disk, coupled to the network, and having persistent storage for data signals, providing plural computers, each having an interface, coupled to the data network, for exchanging data signals between the plural computers, and assigning a portion of the addressable memory space to a portion of the persistent storage of the hard disk to provide addressable persistent storage for data signals.

Detailed Description Text (78):
The private memory device 218 can be any computer memory device suitable for storing data signals representative of computer readable information. The private memory provides the node with local storage that can be kept inaccessible to the other nodes on the network. Typically the private memory device 218 includes a RAM, or a portion of a RAM memory, for temporarily storing data and application programs and for providing the processor 214 with memory storage for executing programs. The private memory device 18 can also include persistent memory storage, typically a hard disk unit or a portion of a hard disk unit, for the persistent storage of data.

Detailed Description Text (84):
FIG. 6 further depicts that the system 230 provides a distributed shared memory that includes persistent storage for portions of the distributed memory. In particular, the depicted embodiment includes a memory subsystem, such as subsystem 232a, that interfaces to a persistent memory device, depicted as the disk 236a. The subsystem 232a can operate the persistent memory device to provide persistent storage for portions of the distributed shared memory space. As illustrated, each persistent memory device 236 depicted in FIG. 6 has a portion of the addressable memory space mapped onto it. For example, device 236a has the portions of the addressable memory space, C.sub.o, C.sub.d, C.sub.g, mapped onto it, and provides persistent storage for data signals stored in those ranges of addresses.

Detailed Description Text (85):
Accordingly, the subsystem 232a can provide integrated control of persistent storage devices and electronic memory to allow the distributed shared memory space to span across both types of storage devices, and to allow portions of the distributed shared memory to move between persistent and electronic memory depending on predetermined conditions, such as recent usage.

Detailed Description Text (102):
The global RAM directory 280 is a directory manager that tracks information that can provide the location of pages that are stored in the volatile memory, typically

RAM, of the network nodes. The global disk directory 284 is a global disk directory manager that manages a directory structure that tracks information that can provide the location of pages that are stored on persistent memory devices. Together, the global RAM directory 280 and the global disk directory 284 provide the shared memory subsystem 270 with integrated directory management for pages that are stored in persistent storage and volatile memory.

Detailed Description Text (104):
The local memory controller of the subsystem 270 is provided by the local RAM cache 276 and the local disk cache 294. The local RAM cache 276 which couples to the physical memory 300 of the local node can access, as described above, the virtual memory space of the local node to access data that is physically stored within the RAM memory 300. Similarly, the local disk cache 294 couples to the persistent storage device 298 and can access a physical location that maintains in the local persistent storage data of the distributed shared memory.

Detailed Description Text (105):
FIG. 8 also depicts a remote operations element 274 that couples between the network 304 and the flow scheduler 272. The remote operations element 274 negotiates the transfer of data across the network 304 for moving portions of the data stored in the shared memory space between the nodes of the network. The remote operations element 274 can also request services from remote peers, i.e. invalidate to help maintain coherency or for other reasons.

Detailed Description Text (108):
The local RAM cache 276 provides storage for memory pages and their attributes. In one embodiment, the local RAM cache 276 provides a global address index for accessing the cached pages of the distributed memory and the attributes based on that page. In this embodiment, the local ram cache 276 provides the index by storing in memory a list of each global address cached in the local RAM. With each listed global address, the index provides a pointer into a buffer memory and to the location of the page data. Optionally, with each listed global address, the index can further provide attribute information including a version tag representative of the version of the data, a dirty bit representative of whether the RAM cached data is a copy of the data held on disk, or whether the RAM cached data has been modified but not yet flushed to disk, a volatile bit to indicate if the page is backed by backing store in persistent memory, and other such attribute information useful for managing the coherency of the stored data.

Detailed Description Text (123):
As shown in FIG. 11, the data associated with the directory pages are distributively stored across the two local memories and duplicate copies can exist. As described above and now illustrated in FIG. 11, the data can move between different local memories and also move, or page, between volatile and persistent storage. The data movement can be responsive to data requests made by memory users like application programs, or by operation of the migration controller described above. As also described above, the movement of data between different memory locations can occur without requiring changes to the directory 340. This is achieved by providing a directory 340 that is decoupled from the physical location of the data by employing a pointer to a responsible node that tracks the data storage location. Accordingly, although the data storage location can change, the responsible node can remain constant, thereby avoiding any need to change the directory 340.

Detailed Description Text (131):
For completeness and by way of definition, a core copy is a copy of a shared page stored on a persistent storage device (e.g., local hard disk of one of the network nodes) that is updated whenever the contents of that page are modified by any network node.

Detailed Description Text (136):
Third, there is a strong bias towards creating loose copies across intercloud links. A "loose copy" of a page (in contrast to a core copy) is a copy stored on a persistent storage device (e.g., a local hard disk of a network node) that is not updated whenever a node modifies the page. To ensure consistency, when a loose copy of a page is activated on a node, its version number is checked against that of any core copy, and if they match, the contents of the loose copy are up-to-date and thus can be served, otherwise the contents of the loose copy are discarded and a new copy of the page's data is loaded from a core copy. Loose copies have good read characteristics, although it may be desirable to aggregate version number checks for blocks of related pages across a slow link. Core copies require flushing on every update across a slow link. This is a bad idea unless the local read rate justifies it. Rather than synchronously updating core copies across slow links, it generally is better to update copies asynchronously in the background if access is frequent enough to warrant doing so.

Detailed Description Text (183):
Normal file system activity will continue (mostly) unhindered during the reconciliation process to reduce the visible impact of reconciliation to users. This is important because reconciliation is potentially a long process if there are a large number of changes that need to be reflected over a slow link. This goal can be accomplished in a number of ways. In general, the file-level locks that the reconciliation process and file system share can be used to avoid serving the contents of a file or directory while the reconciliation process is in the process of reconciling that file or directory. In other words, reconciliation is atomic with respect to normal file access to the same file. If a user attempts to access a file that is not yet reconciled, the old local data is served to the user. Changes to the slave filesystem will be appended to the end of the reconciliation log and need to be handled until reconciliation is complete. As an optimization, it is possible to introduce some form of communication between the file system and the reconciliation process to cause that file or directory to be reconciled synchronously at a high priority so that the file system can serve the most up-to-date data (i.e., shift the lazy reconciliation to a synchronous reconciliation for that file).

CLAIMS:

1. A computer system, comprising:

a first computer network including a first plurality of computers sharing a first globally addressable storage system, each of the first plurality of computers including (a) a local volatile memory device for volatile storage, (b) a local persistent storage device for persistent storage, and (c) a shared memory subsystem for mapping at least a portion of the first globally addressable storage system to a portion or all of the volatile and persistent storage to provide thereby addressable volatile and persistent storage accessible by each of the first plurality of computers; and

a second computer network located remote from and coupled to the first network, the second network including a second plurality of computers sharing a second globally addressable storage system, each of the second plurality of computers including (a) a local volatile memory device for volatile storage, (b) a local persistent storage device for persistent storage, and (c) a shared memory subsystem for mapping at least a portion of the second globally addressable storage system to a portion or all of the volatile and persistent storage to provide thereby addressable volatile and persistent storage accessible by each of the second plurality of computers;

wherein the first and second globally addressable storage systems interoperate to allow the first plurality of computers to access data on the second network including data stored in the local persistent storage devices associated with the

second plurality of computers and to allow the second plurality of computers to access data on the first network including data stored in the local persistent storage devices associated with the first plurality of computers.

14. The computer system of claim 13 wherein the first and second globally addressable storage systems utilize the global directory mechanism which includes a disk directory for tracking data stored on the persistent storage devices and a RAM directory for tracking data stored on the local volatile memory devices on the first and second computer networks.

15. A computer system, comprising:

a computer network; and

a plurality of computers coupled to the network and sharing a globally addressable storage system, at least one of the plurality of computers being located remote from the network and coupled thereto by a communications adapter, each of the plurality of computers including

a local volatile memory device for volatile storage,

a local persistent storage device for persistent storage,

a shared memory subsystem for mapping at least a portion of the globally addressable storage system to a portion or all of the volatile and persistent storage to provide thereby addressable volatile and persistent storage accessible by each of the plurality of computers.

16. The computer system of claim 15 wherein the globally addressable storage system replicates data stored in the local persistent storage devices among two or more of the computers.

17. The computer system of claim 15 wherein the globally addressable storage system replicates data stored in the local persistent storage devices among two or more of the computers based on accesses by the computers of the globally addressable storage system to obtain data stored in the local persistent storage devices.

18. The computer system of claim 15 wherein the globally addressable storage system migrates data stored in the local persistent storage devices among two or more of the computers.

19. The computer system of claim 15 wherein the globally addressable storage system migrates data stored in the local persistent storage devices among two or more of the computers based on accesses by the computers of the globally addressable storage system to obtain data stored in the local persistent data storage devices.

21. The computer system of claim 15 wherein the shared memory subsystem of each of the computers includes:

a distributor for mapping at least a portion of the globally addressable storage system across at least a portion of at least some of the local persistent storage devices to distribute the globally addressable storage system across these local persistent storage devices; and

a disk directory manager for tracking the mapped portions of the globally addressable storage system to provide information representative of which of the local persistent storage devices has which portions of the globally addressable storage system mapped thereon.

Previous Doc Next Doc Go to Doc#

L3: Entry 1 of 9 File: USPT Dec 16, 2003

DOCUMENT-IDENTIFIER: US 6665682 B1
TITLE: Performance of table insertion by using multiple tables or multiple threads

Detailed Description Text (75):
In particular, the relational database 118 locks rows (i.e., using row-level locking) until a commit occurs to avoid having rows modified by a second transaction while a first transaction operates on those rows. A commit is a database operation that indicates that the data in temporary storage, such as a cache, is to be copied to persistent storage, such as a database. Note that the OLAP engine 112 generates a commit based on system settings and client requests and forwards the commit to the relational storage manager 114. When multiple transactions are accessing rows in a single table, the relational database 118 may lock the table to allow only one transaction to operate on the rows of the table. This causes the database to prevent multiple transactions from concurrently accessing a table. Thus, having multiple threads access a single fact table is not efficient.

Detailed Description Text (78):
As shown in FIG. 5, the relational storage manager 500 maintains a cache 510 in memory. Additionally, fact tables 520, non-anchor dimension tables 524, anchor dimension table 526, and key table 516 are stored in persistent storage.

Detailed Description Text (79):
When data is to be written, initially, a data block and its sparse index key are presented to the relational storage manager 500 by a multi-dimensional database calculation engine (MDCE) (which is part of the OLAP engine 112) for writing to persistent storage. The MDCE accesses a single data block at a time for a transaction, with a data block corresponding to data in one fact table. Note that the MDCE actually receives multiple requests from different applications. The MDCE uses a separate thread for each request when communicating with the relational storage manager 500. However, when the MDCE requests data for an application, the MDCE pauses processing for that application until the data is received. After receiving a request from the MDCE, the relational storage manager 500 copies the requested data to a memory-resident cache 510 and returns control to the MDCE.

Detailed Description Text (83):
That is, when the MDCE specifies that a transaction is to be committed, all cache data that has not been written to one of the fact tables 520 is then written to the appropriate fact tables 520 e key table 516, and the RDBMS 116 is instructed to commit all data it has received. Similarly, when there are a predetermined number of "dirty" data blocks in the cache, they are written to the appropriate fact tables 520 and the key table 516.

CLAIMS:

11. The method of claim 10, wherein the base tables are old base tables, and wherein the step of redistributing data further comprising-the steps of: creating new base tables; and moving data from the old base tables into the new base tables using independent threads that use an insert statement with a subselect clause to minimize data movement.

26. The apparatus of claim 25, wherein the base tables are old base tables, and wherein means for redistributing data further comprises: means for creating new base tables; and means for moving data from the old base tables into the new base tables using independent threads that use an insert statement with a subselect clause to minimize data movement.

41. The article of manufacture of claim 40, wherein the base tables are old base tables, and wherein the step of redistributing data further comprising the steps of: creating new base tables; and moving data from the old base tables into the new base tables using independent threads that use an insert statement with a subselect clause to minimize data movement.

Previous Doc Next Doc Go to Doc#